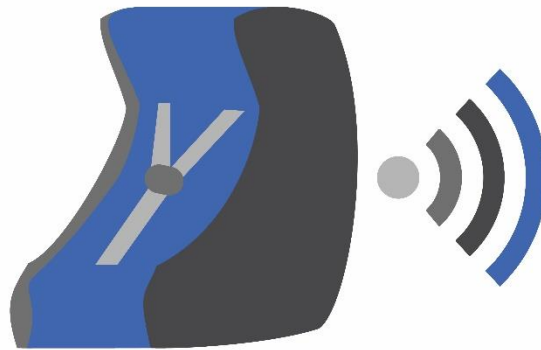


Car Safe

By

David Van Workum



TRAVEL SAFELY

Senior Project

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo

June 2016

Table of Contents

<i>Section</i>	<i>Page</i>
Abstract.....	3
Acknowledgements.....	4
1. Introduction.....	5
2. Customer Needs, Requirements, and Specifications.....	6
3. Functional Decomposition.....	9
4. Project Planning.....	13
5. Project Design.....	18
6. Project Testing.....	25
7. Conclusion and Future Work.....	27
References.....	28
<i>Appendix</i>	
A. ABET Senior Project Analysis.....	30
B. Hardware Schematics.....	34
C. Software Flowchart.....	37
D. Software Code.....	39

List of Tables and Figures

<i>Table</i>	<i>Page</i>
2.1 Car Safe Requirements and Specifications	7
2.2 Car Safe Deliverables	8
3.1 Car Safe Level 0 Functionality	9
3.2 Car Safe Level 1 Functionality	11
4.1 Total Car Safe Time Estimates	16
4.2 Total Car Safe Cost Estimates	16
6.1 Car Safe Specifications and Test Results	26

<i>Figure</i>	
3.1 Car Safe Level 0 Block Diagram	9
3.2 Car Safe Level 1 Block Diagram	10
4.1 Gantt Chart Through EE460	13
4.2 Gantt Chart Through EE461	14
4.3 Gantt Chart Through EE462	15
5.1 Car Safe installed in a generic child seat	18
5.2 Complete Car Safe Transmitter electronics	19
5.3 NRF24L01+ with 10uF capacitor	20
5.4 Car Safe Pressure Sensor	20
5.5 Car Safe Pressure Sensor Interior	21
5.6 Complete Car Safe Receiver electronics	22
5.7 Car Safe Receiver PCB	23
B.1 Car Safe Receiver schematic	34
B.2 Car Safe Transmitter schematic	34
B.3 Adafruit PowerBoost 500c schematic	35
B.4 SparkFun MEMS microphone breakout board schematic	36
C.1 Transmitter (Child Seat Side) program flowchart	37
C.2 Receiver (Parent Side) program flowchart	38

Abstract

The Car Safe project helps protect parents and caretakers from the dangers associated with leaving children in hot vehicles. The device supports parents through notifying them of the child's wellbeing and status while within an automobile. Specifically, the device monitors the child's ergonomic experience and the spatial proximity between the caretaker and child. Car Safe utilizes this information and determines if the child finds himself in a dangerous environment. Due to its compact design, the device functions independently of a specific car seat design, allowing customers to integrate Car Safe in the car seat environment of their choosing.

Acknowledgements

This project could not have happened without the work of a few people. I would like to extend my appreciation towards those who have contributed. First, I would like to thank Dr. Bridget Benson for meeting with me and helping guide me through this project as my senior project adviser. I would also like to thank Dr. David Brawn for helping me throughout the EE460 project planning stage to start this report on the right track. Also, I want to thank my fiancé, Katie Bumpus, for assisting me through testing as well as acting as the videographer for my Car Safe demo video. Finally, I would like to thank my family and friends for their support and encouragement throughout this project and my time at Cal Poly.

Chapter I: Introduction

As parents' schedules become increasingly busy and car usage rises in popularity, the risk of parents leaving their children inside automobiles continually grows. As documented by Jan Null of San Jose State University, in the past decade lawmakers drafted laws protecting children within moving vehicles by increasing car seat regulations [1]. Traditionally, car seats protect their occupants from conventional forms of automobile dangers, primarily car accidents. The protection provided by car seats alone remains necessary but not complete. Since 1998, 661 children in the United States lost their lives in automobiles, not from accidents, but from heatstroke. Of these children, 53% died because their caretakers forgot them inside the vehicle [2]. The Car Safe project aims to reduce this number by providing parents with a system to add "smart" capabilities to the car seat they already own.

The features Car Safe adds to the traditional car seat include: notifications for a parent if he leaves a child inside a car seat unattended, a system monitoring the ergonomic quality of the child's environment, and instant feedback of the child's status. Car Safe accomplishes this by integrating multiple sensors within two small devices. One device remains with the caregiver acting as an information receiver and notification system, and the other device remains with the child functioning as an information transmitter.

Other inventors, influenced by a desire to increase child safety, have developed patents and devices similar to Car Safe. Two such patents are an alarm child car seat filed by Robert Lemons and a child safety seat mobile alarm filed by Amy Rambadt. The first operates by sounding an alarm when a pressure sensor within the car seat remains activated long after a vehicle's engine turns off [3]. The latter operates similarly to Car Safe by utilizing a sensing device external to the car seat itself to send notifications to users when the user distances him or herself from the device. The device transmits this information via SMS, E-mail, or social media [4]. Car Safe differentiates itself from other devices through its focus on providing feedback to caretakers beyond notifying them of a forgotten child.

Chapter II: Customer Needs, Requirements, and Specifications

Customer Needs Assessment

The Car Safe customer base primarily encompasses young parents. The characterization of this demographic includes families with low to mid-level incomes, busy schedules, and young children. With this in mind, this customer needs assessment developed. Personal interviews with six young parents in Orange County, CA further confirmed the customer needs that are previously determined. The questions asked during the interviews included:

1. If you were to utilize a car seat with “smart” capabilities, what type of information would you want to learn from it?
2. What are the daily difficulties and inconveniences you experience using a car seat?
3. What concerns do you have when considering purchasing a device that adds “smart” capabilities to a car seat?

These interviews show that customers prioritize four basic product needs: ease of use and assembly, child comfort and safety, product durability, and product affordability [5].

Requirements and Specifications

Car Safe’s requirements and specifications developed from the previous customer needs assessment. The expected busy life of a young parent requires devices that interact seamlessly with their day-to-day life. The customer desires to install and set up Car Safe quickly and easily each day. As the use model of Car Safe includes both day-to-day commutes and long road trips, the device must accommodate the ergonomic and power needs of each trip. These needs promote requirements for device comfort, device durability, and long battery life. Finally, as Car Safe promotes child safety as its primary function, it must comply with child safety standards. These standards include, but are not limited to: chemical safety, thermal safety, and physical safety standards. A complete list of Car Safe’s marketing requirements and engineering specifications appears in Table 2.1. Due to limited funding for Car Safe, the development costs should not surpass \$200.

Table 2.1 Car Safe Requirements and Specifications

Spec #	Marketing Requirements	Engineering Specifications	Justification
1	3, 4	Product contains less than 100 ppm of certain phthalates including DEHP, DBP, and BBP.	Eliminating phthalates ensures a chemically safe product, preventing sickness and skin irritation. This phthalate level remains within the United States Consumer Product Safety Commission's standard for child toys [6].
2	1	An average adult parent can set up and install the product in under 10 minutes.	A short setup and installation supports an easy installation claim.
3	2	Product withstands 50 consecutive drops from 10 ft.	The device remains intact and functional when dropped from the car.
4	5	The device's battery holds charge for at least 24 hrs. during constant use.	The product operates without recharging for the duration of a long road trip.
5	3, 4	A child placed in a car seat has the same range of motion with and without the device in place.	The device does not impede on the user's car seat experience.
6	4, 5	The product's temperature remains under 100 degrees F during a 24 hr. constant use period in a 70 degrees F environment.	The device remains within a comfortable temperature to the human touch, supporting low energy consumption and safe operating environments [14].
7	4	The volume of the child side device exceeds 2 in. ³ .	The product should not fit within a Choke Test Cylinder 1.25 in. wide and 2.25 in. deep [7].
8	6	The cost for consumer does not exceed \$250.	The device's price point sits in a competitive location amongst smart car seat devices.
9	7	Transmitting device and receiving device operate within 50 ft. of each other when unobstructed.	This provides a buffer so that a "forgotten child" notification only occurs when a caregiver walks a significant distance from the transmitter.
10	7	Emergency notifications (child unattended, high temperature, high noise level) presented within 1 second of emergency occurrence.	Caretakers receive notifications without initializing a status assessing sequence and within a proper time to attend to the problem.

11	3, 4	Device measures the audio level of signals within 80 Hz to 20 kHz, up to 100 decibels within 10% accuracy.	Notifications occur when the sound a baby experiences, while near Car Safe, rises beyond a comfortable volume of 80 decibels [11].
	Marketing Requirements <ol style="list-style-type: none"> 1. Easy Installation 2. Durable 3. Comfortable 4. Child Safe 5. Long Battery Life 6. Low Cost 7. Non-Intrusive 		

Specified project deliverable appear in Table 2.2.

Table 2.2 Car Safe Project Deliverables

Delivery Date	Deliverable Description
Nov. 5, 2015	ABET Sr. Project Analysis
Feb. 25, 2016	EE 461 Pre-alpha Demo
Feb. 23, 2016	EE 461 Alpha Demo
Apr. 28, 2016	EE 462 Beta Demo
Apr. 28, 2016	EE 462 Report Rough Draft
May 27, 2016	College Wide Senior Project Expo
Jun. 8, 2016	Final Report

Chapter III: Functional Decomposition

Level 0 Decomposition

The Car Safe device accepts an array of information and outputs feedback based upon this information. Figure 3.1 presents the device's level 0 black box diagram. The inputs and outputs shown represent abstract information transmission rather than electrical signal transmission. Table 3.1 further explains the inputs and outputs within the diagram. It also describes the functionality of the box itself.



Figure 3.1: Car Safe Level 0 Black Box Diagram

Table 3.1 Car Safe Level 0 Functionality

Module	Signal Description
Inputs	Heat - Expected temperature range of 0°F to 120°F. Audible Sound - Expected frequency range of 80Hz to 20kHz. User Proximity - Expected proximity range of 0m to 100m. Human Baby Presence - Expected weight range of 5lbs to 30lbs and expected height range of 19in to 35in. USB Power 5V- Power supply for circuit. Power Switch - Physical on/off switch.
Outputs	Audio Notification- Produced by Piezo Buzzer. Visual Notification- Produced by LED.
Functionality	The Car Safe Device utilizes an array of sensors that present valuable information about the status of a child in a car seat. It takes measurements based on its 3 primary inputs: heat, user proximity, and human baby presence. It uses the information it collects to notify a parent both audibly and visually if a child faces possible neglect. Device activation occurs when the user moves the power switch from the off to the on position. By closing this switch, the user allows power to flow from 5V USB source to the car safe device.

Level 1 Decomposition

The level 1 box diagram presented in Figure 3.2 further deconstructs the information shown in the previous figure. As seen in the figure, an array of sensing devices measure the abstract information input to the device. These devices pass the translated information to a processing device that analyzes the signals. Finally, the processing device transmits commands controlling the devices output. Table 3.2 goes into greater detail on Car Safe's level 1 functionality.

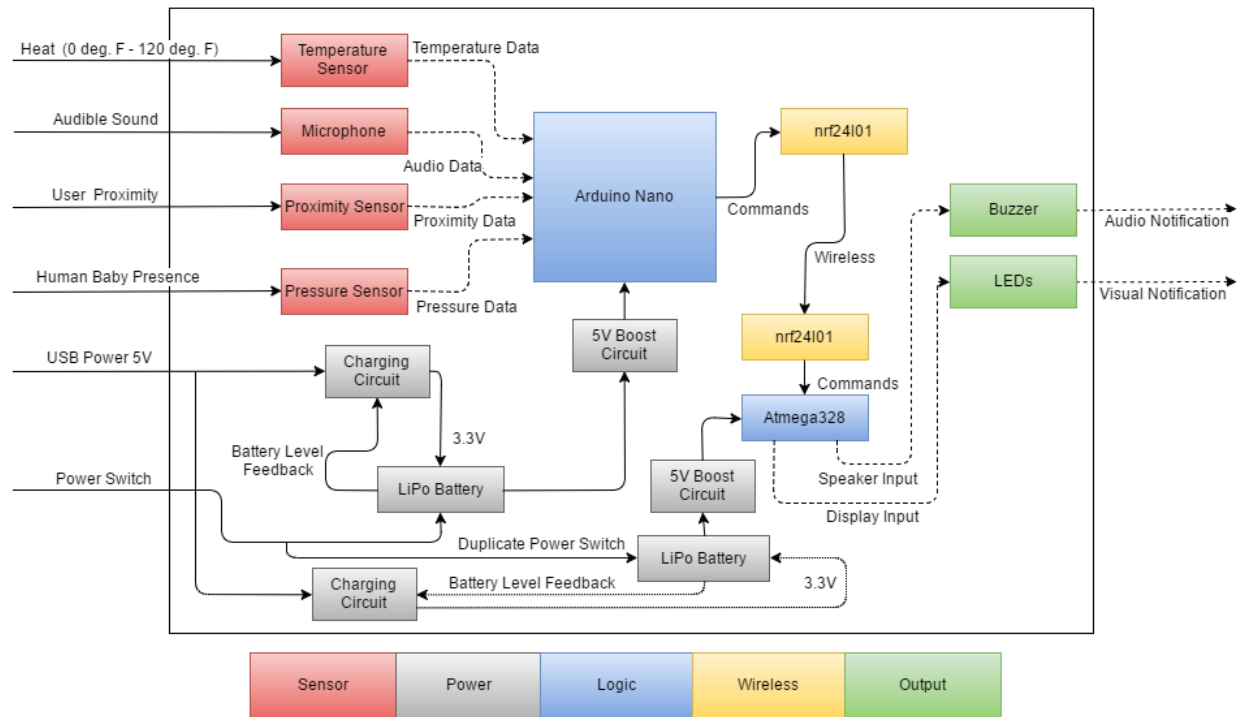


Figure 3.2: Car Safe Level 1 Black Box Diagram

Table 3.2 Car Safe Level 1 Functionality

Charging Circuit	Inputs	- 5V, 1A DC electricity from a USB supply - Battery voltage level
	Outputs	- 5V DC
	Functionality	When 5V appears on its input, the circuit regulates the charge held by the 5V rechargeable battery.
5V Battery	Inputs	- 5V DC from the charging circuit - Power switch position
	Outputs	- 5V, 1A DC electricity from a USB supply - Battery voltage level
	Functionality	The battery recharges via the charging circuit attempting to retain a 5V charge. The power switch input controls when the battery outputs 5V. If the switch remains in the “off” position, no power flows to other circuits.
Temperature Sensor	Inputs	- Heat from the environment
	Outputs	- Temperature measurement information
	Functionality	The temperature sensor converts heat into quantifiable information, temperature, and sends the information to the microcontroller for processing.
Microphone	Inputs	- Audible sound within a frequency range of 80Hz to 20kHz
	Outputs	- Audio decibel and frequency information
	Functionality	The microphone receives an audio signal and converts it to an electrical signal readable by a microcontroller.
Proximity Sensor	Inputs	- Proximity of transmitter and receiver
	Outputs	- User proximity information
	Functionality	The proximity sensor measures if a user holding an NRF24I01 receiver remains within a specified distance from the transmitter. It then outputs this information to the microcontroller.
Pressure Sensor	Inputs	- Presence of baby
	Outputs	- Baby presence information
	Functionality	The pressure sensor examines whether or not a human baby appears within the confines of the Car Safe device. If it finds this true, it notifies the microcontroller.

Microcontroller	Inputs	<ul style="list-style-type: none"> - 5V - Temperature measurement information - Audio decibel and frequency information - User proximity information - Baby presence information
	Outputs	- Wireless Bluetooth transmission
	Functionality	The microcontroller acts as Car Safe's central hub. It analyzes the information it receives from the various centers and decides whether or not it activates the device speaker and display. It transmits this data wirelessly via an internal Bluetooth transmitter.
Bluetooth Receiver	Inputs	- Wireless signal from microcontroller
	Outputs	<ul style="list-style-type: none"> - Speaker control signal - Display control signal
	Functionality	The Bluetooth receiver receives information from the microcontroller containing the necessary commands pertaining to the device speaker and display and transmits them accordingly.
Piezo Buzzer	Inputs	- Speaker control signal
	Outputs	- Audible Car Safe status signal
	Functionality	The Buzzer activates when encouraged by the radio receiver and microcontroller. It acts as an alarm to notify a user when the baby experiences an unsafe environment.
Display	Inputs	- Display control signal
	Outputs	- Visual Car Safe status signal
	Functionality	The display activates when the Bluetooth receiver and microcontroller functions as a visual indicator of the state of the child and device.

Chapter IV: Project Planning

The project plan formed at the beginning of the Car Safe project assists in providing organization and accountability for the project. When planning a project, two main resources require accounting. These resources appear as time and money. Figure 4.1, Figure 4.2, and Figure 4.3 show a singular Gantt chart creates an expected timeline for the project. The chart exists, split into three figures for the reader's convenience.

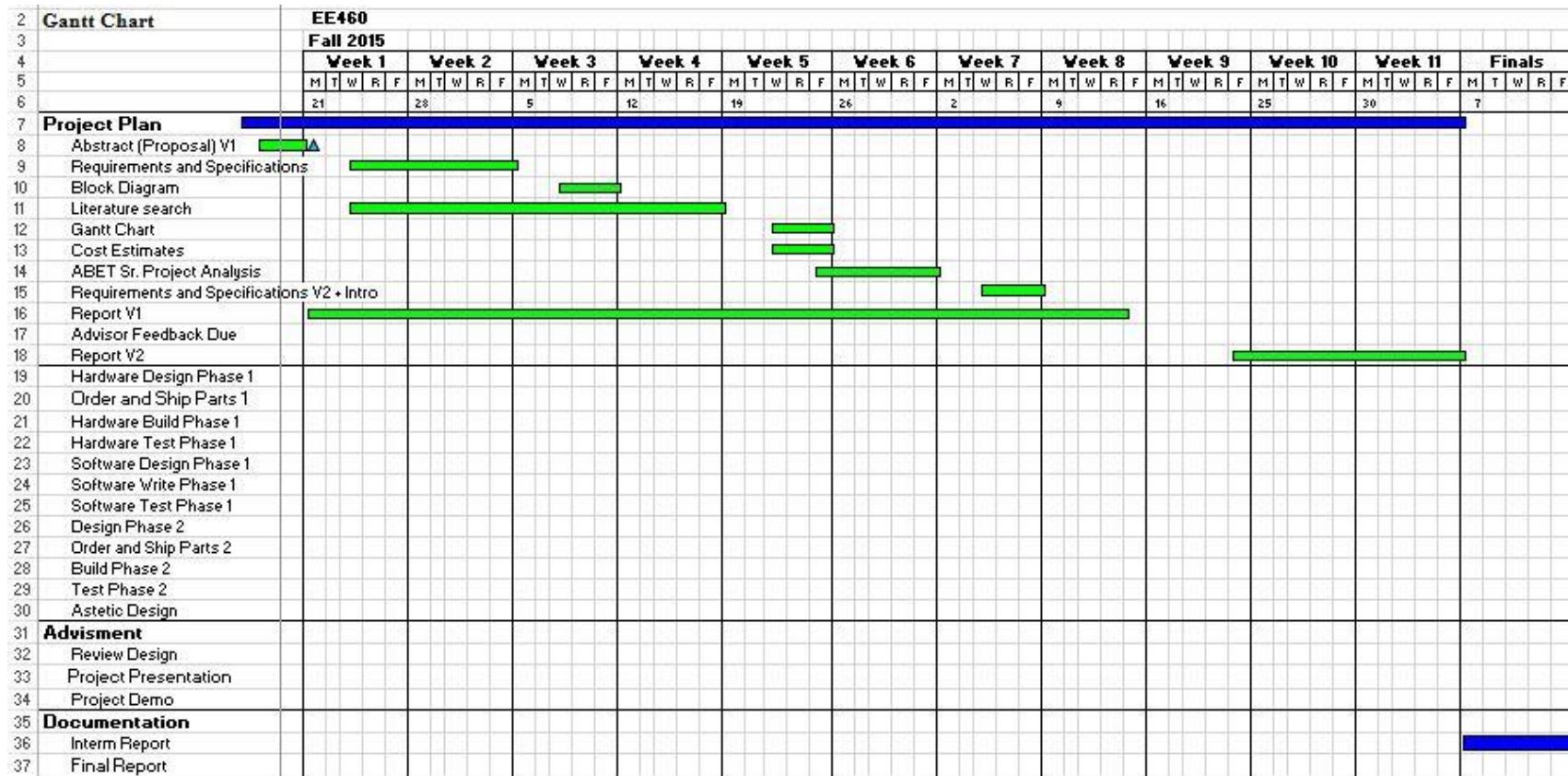


Figure 4.1: Gantt chart through EE460

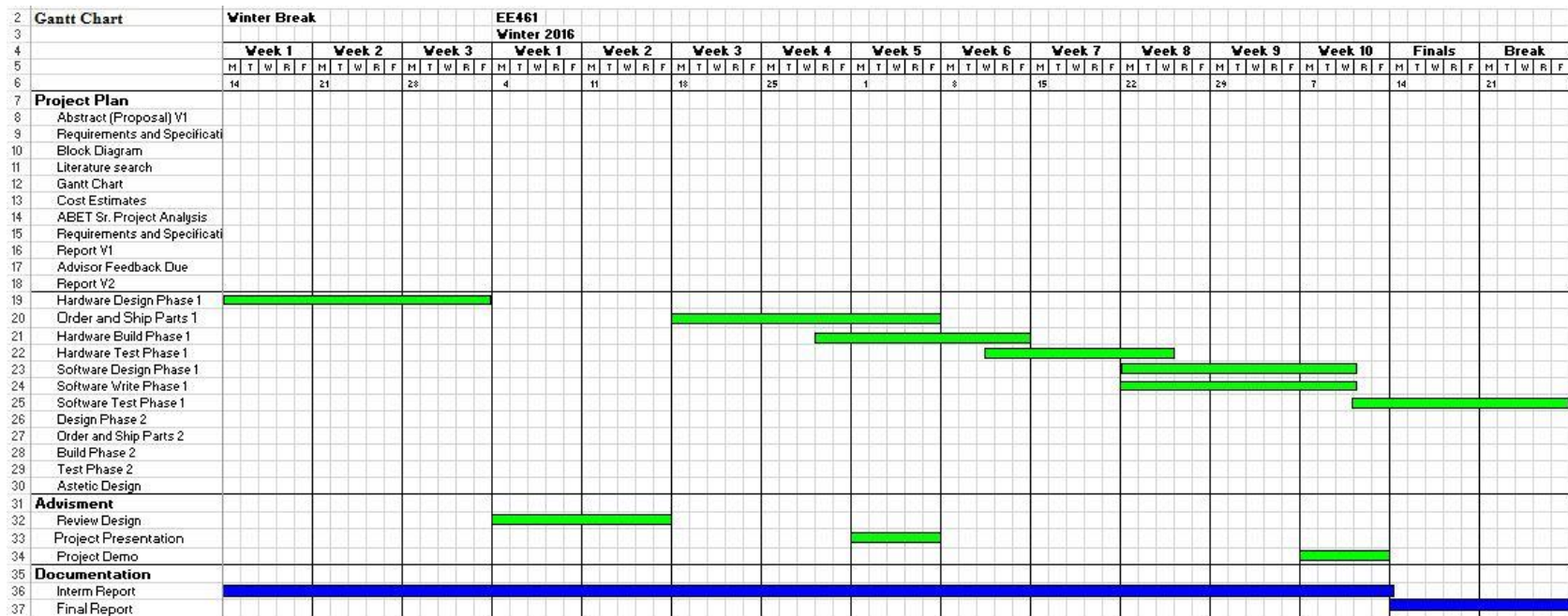


Figure 4.2: Gantt chart through EE461

Gantt Chart		EE 462 Spring 2016																																																							
		Week 1					Week 2					Week 3					Week 4					Week 5					Week 6					Week 7					Week 8					Week 9					Week 10					Finals					
		M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F						
		28					4					11					18					25					2					9					16					23					30					6					
Project Plan																																																									
Abstract (Proposal) V1																																																									
Requirements and Specifications																																																									
Block Diagram																																																									
Literature search																																																									
Gantt Chart																																																									
Cost Estimates																																																									
ABET Sr. Project Analysis																																																									
Requirements and Specifications																																																									
Report V1																																																									
Advisor Feedback Due																																																									
Report V2																																																									
Hardware Design Phase 1																																																									
Order and Ship Parts 1																																																									
Hardware Build Phase 1																																																									
Hardware Test Phase 1																																																									
Software Design Phase 1																																																									
Software Write Phase 1																																																									
Software Test Phase 1																																																									
Design Phase 2																																																									
Order and Ship Parts 2																																																									
Build Phase 2																																																									
Test Phase 2																																																									
Aesthetic Design																																																									
Advisement																																																									
Review Design																																																									
Project Presentation																																																									
Project Demo																																																									
Documentation																																																									
Interim Report																																																									
Final Report																																																									

Many hours go into Car Safe's design and manufacturing. These hours account for the design, build, and testing phases of the device's hardware and software, as well as the documentation work placed into the device. Table 4.1 lays out the hours associated with each task, resulting in a total time exhausted of 238.85 hours. One notes that the time shown for task completion within the Gantt chart appears longer than the hours represented in Table 4.1. This allows for the splitting of hourly labor time into multiple days. Each time estimate derived for the table relies upon the PERT method of estimating [8]. The following equation shows the method's implementation:

$$T_{estimated} = \frac{T_{optimistic} + 4T_{realistic} + T_{pessimistic}}{6} \quad \text{eq. 1}$$

Table 4.1 Total Car Safe Time Estimate

Category	Task	Time (hr.)
Research & Development	Hardware Design	60
	Software Design and Writing	60
	Project Planning Documentation	55
	Final Project Documentation	25.83
	Total	200.83
Manufacturing and Technician Work	Hardware Building	35
	Hardware Testing	25.72
	Software Testing	22.3
	Total	83.02
Total		283.85

The Car Safe project's costs arise from a combination of labor and component costs. Table 4.2 presents a breakdown of these costs. Research and Development costs represent the largest fraction of the total project cost. One can consider this opportunity cost since the dollar figure represents the wage the project developer relinquished by spending time working on his senior project instead of working as a Research and Development intern at an established technology company [9].

Table 4.2 Total Car Safe Cost Estimates

	Item	Cost/Unit	Quantity	Total	Explanation
Labor Cost	Research & Development	\$30/hr.	200.83 hr.	\$6,024.90	The labor costs accrued from time spent developing and documenting the Car Safe project. An hourly wage of \$30/hr. applied to the time represents the average wage paid to an intern within an R&D position.
	Manufacturing and Technician Work	\$20/hr.	83.02 hr.	\$1,660.40	The labor costs accrued from time spent building and testing the Car Safe project. An hourly wage of \$20/hr. applied to the time represents the average wage paid to an intern within a manufacturing position.

Part Cost	Arduino Nano	\$2.27	1 unit	\$2.27	
	MEMS Microphone	\$9.95	1 unit	\$9.95	
	NRF24L01+ Radio Transceiver	\$3.23	2 units	\$6.46	
	Temperature Sensor	\$1.50	1 unit	\$1.50	
	1200mAh LiPo Battery	\$9.95	2 units	\$19.90	
	PowerBoost 500 Charger	\$14.95	2 units	\$29.90	
	Piezo Buzzer	\$3.00	1 unit	\$3.00	
	Osh Park PCB	\$6.00	1 unit	\$6.00	
	Miscellaneous Discrete Components and Parts	\$40	1 unit	\$40	Expected Cost of additional components needed for the device including resistors, capacitors, diodes, transistors, and operational amplifiers.
Total Cost				\$7,804.28*	

*Component costs do not for shipping, tax, or expedited production costs

Chapter V: Project Design



Figure 5.1: Car Safe installed in a generic child seat

Design Concept

Car Safe's design is centered on the co-utilization of hardware and software. With two microcontrollers at its core, the device relays information regarding the status of the child's wellbeing between its transmitting (child seat side) and receiving (parent side) components. This information is gathered via a pressure sensor, a temperature sensor, and a volume sensor located on the child seat side of the device. Two identical radio transceivers then wirelessly transmit the data to the parent side device. With the child's status in hand, the parent side device notifies the user via LED's and a piezo buzzer if the child is in danger. Above, figure 5.1 demonstrates what Car Safe looks like when installed within a child's car seat.

Hardware Design

Child Seat Side



Figure 5.2: Complete Car Safe Transmitter electronics

Microcontroller

At the core of the child seat side of the device sits an Arduino Nano microcontroller. This Arduino acts as the primary source of control for Car Safe. The Nano's small size and easy accessibility made it very easy to work with during Car Safe's development. Since it maintains the same functionality as a full-fledged Arduino Uno, it can be programmed using the Arduino Sketch ecosystem and all of the libraries associated with an Arduino Uno. Because of this, Car Safe makes use of a transceiver pair that often associates with an Arduino Uno [16]. The Arduino can be seen in the center of Figure 5.2 above.

Wireless Transceiver

Car Safe inherently acts as a wireless communication device. The device must notify a parent standing away from a child seat if they leave the child in the car. To perform this notification task, Car Safe uses the wireless transmitter's own transmitting range limitations to its advantage. When the receiver fails to receive a signal from the transmitter while a child remains seated in a car seat, the receiver notifies the user that he has forgotten his or her child inside the car.

In selecting a transceiver pair, the following criteria was considered:

- The device should be low cost.
- The device should have limited range.
- The device should be easily implementable.

While other radio technologies including Bluetooth and Wi-Fi were considered, the Nordic Semiconductor's NRF24L01+ proved the best fit for the project. The 2Mbps transceiver features a relatively low peak RX/TX current draw of 14mA and integrates easily with an Arduino through the SPI communications protocol [17].

During testing, output current limitations associated with the Arduino 3.3V output caused a supply voltage stability problem for the NRF24L01+. A 10uF capacitor that was soldered between the ground and supply terminals on each transceiver remedied this problem. This can be seen in figure 5.3.

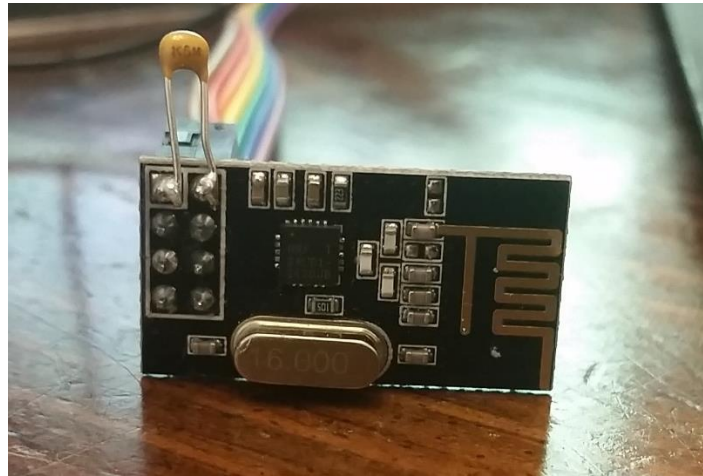


Figure 5.3: NRF24L01+ with 10uF capacitor

Sensors

The transmitting side of Car Safe consists of three sensors: a pressure sensor, temperature sensor, and a noise level meter. The pressure sensor holds a very important role within the Car Safe system. The device indicates whether or not a child is currently seated in a car seat. This information remains pertinent since a parent should not receive an emergency notification if a child is away from the car seat. When choosing a pressure sensor, very few sensors available in the market proved acceptable for the project. Resistive touch sensors are too sensitive. Load cells, on the other hand, such as those found in commercially available bathroom scales, are less sensitive but need additional hardware that proved costly and ineffective. Instead, Car Safe utilizes a custom sensor built to act as a momentary switch shown in Figure 5.4 below.

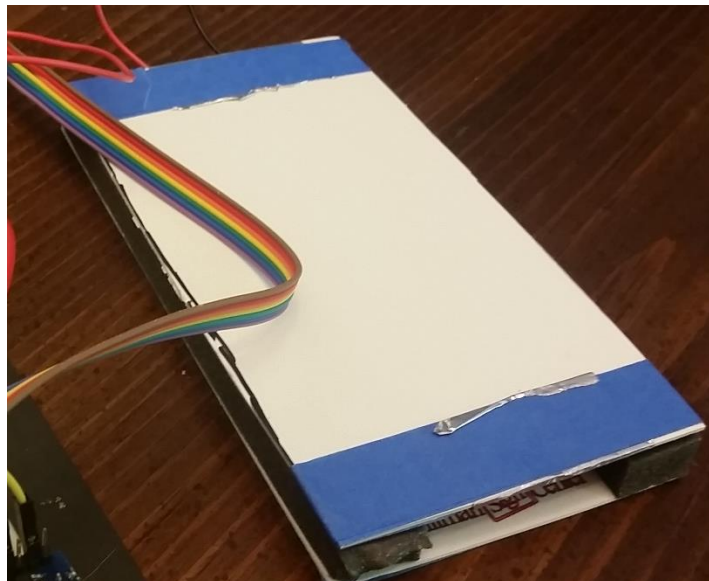


Figure 5.4: Car Safe pressure sensor

The switch's construction consists of two sheets of thin plastic separated by $\frac{3}{4}$ inches of foam. Two metallic conductors secured within the two plastic sheets make contact, creating a short circuit with the application of pressure to the sensor. The internal structure of the device can be seen in Figure 5.5 below.

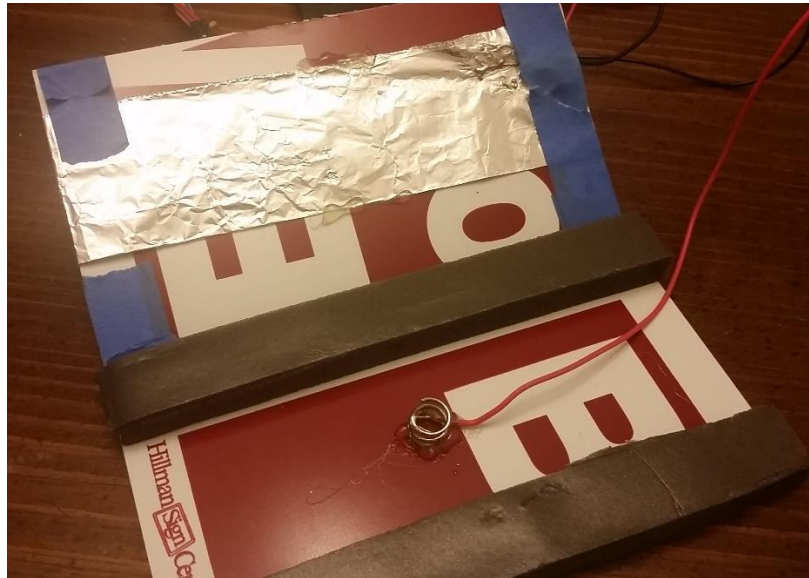


Figure 5.5: Car Safe pressure sensor interior

The sensor connects to the Arduino in the same way that a common push button would connect, causing an Arduino pin to pull high when pressed.

To measure the thermal temperature experienced by the child, Car Safe utilizes Analog Device's TMP36 Low Voltage Temperature Sensor. The component itself operates between -40°C and 125°C , adjusting its output voltage at a rate of $10\text{ mV}/^{\circ}\text{C}$. The Arduino's internal ADC reads the output and determines the temperature [18].

Car Safe's noise level monitor utilizes the MEMS Microphone Breakout board from SparkFun. The microphone picks up audio from 100Hz to 15kHz. The breakout board itself features an OPA344 op-amp that amplifies the signal to a level that the Arduino's ADC can distinguish. The schematic for the breakout board is in Appendix B.

Power

Car Safe's transmitter and receiver utilize identical power supplying circuits. Each employs a 3.7V, 1200mAh Lithium Polymer battery as an energy reserve. In order to implement the batteries within the Car Safe ecosystem, two things must happen. First, the batteries must charge. Second, the 3.7V must step up to 5V. To accomplish these two requirements, Car Safe contains the PowerBoost 500c board from Adafruit. This board consists of a charging circuit with a micro USB connector and a 5V boost converter. A complete schematic of the PowerBoost 500c can be found in Appendix B.

Parent Side

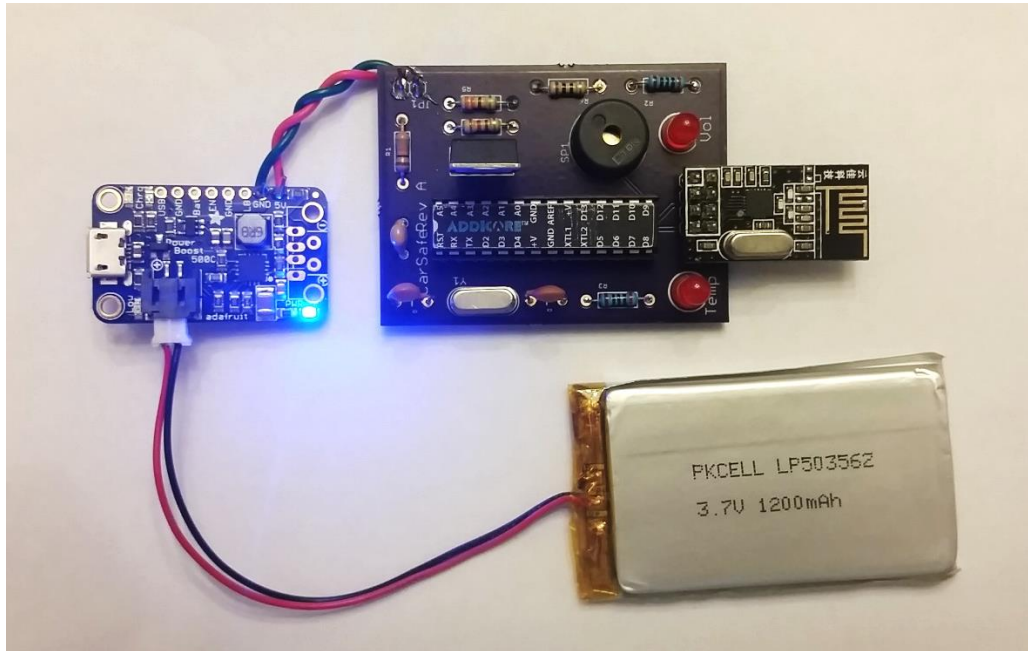


Figure 5.6: Complete Car Safe Receiver electronics

Microcontroller

The receiver side of the Car Safe device utilizes an Atmega328P, which is the same chip used in the Arduino Nano. Since the Car Safe receiver must incorporate a mobile, lightweight design, the Atmega328P was removed from the Arduino and placed in custom board that houses the parent side components. The custom PCB contains components that allow the Atmega328P to run as if it were within an Arduino Uno. These components include a 16 Mhz clock crystal, pull-up resistor, and 22 pF capacitors. With these components in place the NRF24L01+ utilize the same code libraries used by the Car Safe transmitter. While the Atmega328P at the receiver may run the same code as the Arduino Nano used by the transmitter, the receiver board does not feature a convenient way to upload code updates. To do so, the chip must be removed from the board and placed within an Arduino Uno. While inconvenient, this keeps the receiver board small since an FTDI chip and USB adapter are not needed.

Notification Components

In addition to the components needed to allow the Atmega328P to function as if it were within an Arduino environment, the Car Safe receiver features components used to notify the parent if an emergency state has been triggered. A piezo buzzer from TDK functions as Car Safe's primary source of audio notification. The device itself vibrates at the frequency of a square wave it receives, resulting in an audible noise [19]. Using the Atmega328P to send the buzzer, a sequence of frequencies creates various alarms that can be associated with different emergency states. As a visual notification, Car Safe contains two red LEDs, which light up when a corresponding emergency state gets triggered.

A complete schematic of Car Safe can be seen in Appendix B.

PCB Design

The design of the custom PCB took place within the Eagle design environment. In order to develop Car Safe's schematics and board layout, third party component libraries created by SparkFun were implemented [20]. The PCB's principle design goal was to facilitate an all-inclusive board space for all of

the receiver side components. To do so, the board features two sets of header pins, allowing the NRF24L01+ transceiver board and the PowerBoost 500c to connect directly to the PCB. Figure 5.6 above displays the final board design.

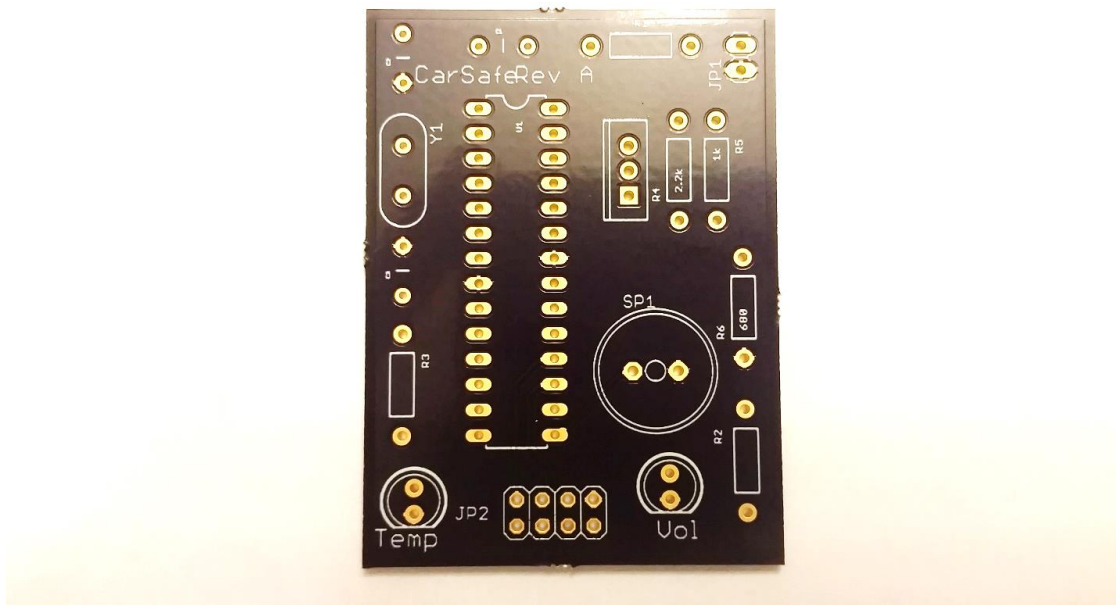


Figure 5.7: Car Safe receiver PCB

The board itself consists of two layers with most of the traces on the bottom layer. The top layer contains a ground plane to keep a stable, consistent ground reference between the components. It also limits any potential signal integrity issues associated with the wireless receiver. The PCB implements copper pads for thru-hole components. Future revisions should likely feature a surface-mount component design to facilitate a smaller board. Figure 5.7 shows an image of the top side of the receiver board without soldered components.

Software Design

Like Car Safe's hardware design, the software design can be divided into two functions: transmitting and receiving. The programming of each microcontroller took place within the Arduino IDE and fully utilized the associated libraries available within the Arduino environment. To write the code used to communicate with the two NRF24L01+ transceivers, the software design implements an external `rf24.h` library from TMRh20 [21]. This library helps facilitate all SPI communication between the Arduino and transceiver.

On power up, the transmitter side begins an initialization process to prepare to read data from its sensors and send information via the transceiver. Once each component is properly initialized, the device enters its main loop. Here, each sensor level gets checked periodically. A chain of if-then statements determines whether or not the sensor level exceeds acceptable limits. This result, along with the status of the pressure switch, is wrapped in a package for the NRF24L01+ to send to its counterpart.

After initializing its own NRF24L01+, the Atmega328P on the parent side waits for the transceiver to receive the package. It then copies this data and examines it. Finally, it determines if it needs to activate the notification and alarm devices because of an emergency state.

The receiver side code determines if an emergency state notification can occur based on the previous data it collected. Specifically, it checks to see whether or not a child sat on the pressure sensor during the last

transmission. If a child was not present when the transmitter sent an emergency notification, the microcontroller will not activate the buzzer or LEDs.

The controller continues in this loop until a radio signal fails to appear. If this occurs for 500 code cycles, and the previous pressure sensor state was high, a state of emergency is entered as the parent has left their child in the car seat.

Throughout both the transmitter and receiver code, calls to the Arduino Sketch serial monitor appear. These calls exist primarily for debugging purposes. In addition to debugging, the information allows for determining the ADC output levels that occur when various temperature and volume levels generate voltage output by their respective Car Safe sensors.

Program flowcharts for each unit appear in Appendix C. The complete code for each program can be found in Appendix D.

Chapter VI: Testing

The testing of Car Safe centers on isolating the various components of the device as well as the product as a whole. Initially, the quality of the data transmission between the child seat and parent side devices proved inconsistent. To isolate the problem, the transceivers were tested by running a simple program that sent a single integer value between the two devices. When the problem persisted, an external 3.3V supply capable of higher supply current levels than that of the Arduino was implemented. This confirmed the suspected problems the NRF24L01+ might have with using the Arduino as a supply and led to the addition of stabilizing 10uF capacitors.

The transition from a breadboard design to a PCB presented some initial challenges. While the transmitter side utilizing the Arduino Nano continued to transmit data as expected, the receiver failed to capture the data and acknowledge the transmitter. After verifying that the traces on the PCB made connections as expected, the NRF24L01+ receiver was removed and placed back in a breadboard to verify that no damage had occurred to the unit when soldered. Once verified, the unit was soldered to the PCB once again. The output voltage levels of the 5V boost converter and 3.3V voltage regulator were then measured. The output of the 3.3V regulator appeared significantly lower than the 3.3V measured when within a breadboard. It was determined that the NRF24L01+ was attempting to draw more current than the regulator could supply. Redesigning the regulator circuit by implementing smaller resistor values alleviated this problem. This change is represented in the schematic within Appendix B, but is not represented in the Rev A PCB silk screen used within this project.

After the completion of the Car Safe design and build process, tests were performed to ensure the device's adherence to the list of Requirements and Specifications shown in Table 2.1. As shown below, Table 6.1 offers the results of those tests.

Table 6.1 Car Safe Specifications and Test Results

Specification	Result	Test Description
1	Not Tested	At Car Safe's current build state, testing for the chemical safeness of the products enclosure would be fruitless since a permanent, production ready enclosure has not yet been designed.
2	Pass	Installation requires placing a stick-on Velcro strip on any free location on a child seat and placing the pressure sensor in the area the child sits. The measured install time is much faster than 10 minutes.
3	Not Tested	At Car Safe's current build state, consecutive drop testing is ill-advised. More development on a permanent, production ready enclosure is required to pass this test.
4	Pass	The Car Safe device was left on for a 24 hr. period. Each device remained within transmitting distance simulating the distance between the front and backseat of a car.
5	Pass	A life sized doll placed in the child seat with and without Car Safe experiences the same level of mobility.
6	Pass*	During the testing of specification 4, the temperature notification failed to activate.
7	Pass**	The child side device in its current state has dimensions of 2.5 in. X 4 in. X 2 in. and does not fit through a Choke Test Cylinder.
8	Pass	The cost of the device within a manufacturing setting falls far below \$100. Therefore, a price point not exceeding \$250 is reasonable for profit.
9	Pass	With the transmitting device stationary, the receiving device was moved away until the signal was lost and the buzzer sounded. The dropout distance was measured to exceed 50 ft. with unobstructed transceivers. When transmitting between walls, however, the dropout distance occurs at about 30 ft.
10	Pass	The child seat side device was placed in a state of emergency and a timer was activated to measure the response of the piezo buzzer on the parent side device. The buzzer consistently activated in under 1 second.
11	Pass	Various 85+dB audible signals were placed within the environment of Car Safe. This consistently activates Car Safe's "loud volume" notification. Signals below 80dB do not activate this notification.

* The temperature of the test environment was not controlled and constant.

** The test must be performed again when a final production ready enclosure is designed

Chapter VII: Conclusion and Future Work

As it stands now, Car Safe works well as a proof of concept, however, improvements must be made to consider it a marketable product. The device succeeds in meeting the basic functionality as described in the requirements section of this report. Car Safe successfully notifies a parent if they walk away from a child still seated within a child seat as well as alerts them when their child is experiencing a potential discomfort in relation to the temperature or sound level of its environment. While the device remains functional, it is not currently practical.

In order to achieve device practicality, Car Safe must exist in a smaller package. While smaller than the transmitter, the receiver PCB, battery, and charging components take up too much space for the average person to conveniently carry around in his or her pocket or on a keychain. In the future, replacing the battery and microcontroller with slimmer options could improve the overall size of the device.

On the transmitter side, the pressure sensor material and design should also improve. Lessening the thickness of the sensor would increase the device's overall comfort as well as support its ability to fit within more car seats. The sensor and wires that connect it to the rest of the transmitting device could also be covered with fabric that adds a professional feel and look to the product.

References

- [1] J. Null, "Heatstroke Deaths of Children in Vehicles," *noheatstroke.org*, Oct. 29, 2015. [Online]. Available: <http://noheatstroke.org/>. [Accessed: Nov. 9, 2015].
- [2] "Child Safety Seats," *chp.ca.gov*, [Online]. Available: <https://www.chp.ca.gov/programs-services/programs/child-safety-seats>. [Accessed: Nov. 9, 2015].
- [3] Lemons, Robert (Mission, TX), 'Alarm child car seat', 9139128, 2015.
- [4] Rambadt, Amy, Rambadt, John, Tanner, David, Kehler, Ken, Strickland, Joseph D. (Gravette, AR), 'Child Safety Seat Mobile Alarm and Method Therefor', 0253314, 2014.
- [5] D. Van Workum, "Car Safe Potential Customer Interviews," Interview, San Luis Obispo, CA, 2015.
- [6] "Phthalates" *cpsc.gov*. July 7, 2015. [Online]. Available: <http://www.cpsc.gov/en/Business--Manufacturing/Business-Education/Business-Guidance/Phthalates-Information/>. [Accessed: Nov. 11, 2015].
- [7] "Avoiding Choking Hazards in Child's Toys," *childsafetycentral.com*, 2006. [Online]. Available: <http://childsafetycentral.com/toy-choking-hazards.html>. [Accessed: Nov.11, 2015].
- [8] R.M. Ford and C.S. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2007.
- [9] A. Ertas, J. C. Jones, *The Engineering Design Process*. New York: John Wiley & Sons, Inc., 1993.
- [10] "E-waste is the Toxic Legacy of our Digital Age," *ifixit.org*. [Online]. Available: <http://ifixit.org/ewaste>. [Accessed: Nov. 11, 2015].
- [11] S. Brangui, M. E. Kihal, Y. Salih-Alj, "An Ergonomic Comprehensive System for Monitoring and Control of Baby Environments," in 2015 6th International Conference on Information and Communication Systems, Ifrane, Morocco. Available: IEEE Xplore, <http://ieeexplore.ieee.org>. [Accessed Oct. 19, 2015]
- [12] E. Bayaki, L. Lampe, R. Schober, "Performance Evaluation of Bluetooth Systems with LDI, Modified LDI, and NSD Receivers," *IEEE Transactions on Vehicular Technology*, vol.57, no.1, January 2008. [Online]. Available: IEEE Xplore, <http://ieeexplore.ieee.org>. [Accessed Oct. 19, 2015]
- [13] "Bluetooth," *Wikipedia.com*, Oct. 15, 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Bluetooth>. [Accessed: Oct. 19, 2015].
- [14] K. H. E. Kroemer, H.B. Kroemer, K. E. Kroemer-Elbert, *Ergonomics: How to Design for Ease and Efficiency*. New Jersey: Prentice-Hall, 1994
- [15] Institute of Electrical and Electronics Engineers, Inc., IEEE Policies, New York, N.Y. IEEE, 2015. Available: https://www.ieee.org/documents/ieee_policies.pdf [Accessed Nov. 12, 2013] pp. 7.3.
- [16] "ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH: DATASHEET," 2014. [Online]. http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf. [Accessed: Oct. 19, 2015].

[17] “nRF24L01+ Single Chip 2.4Ghz Transceiver Product Specification,” 2008. [Online]. http://www.nordicsemi.com/eng/content/download/2726/34069/file/nRF24L01P_Product_Specification_1_0.pdf [Accessed May 1, 2016].

[18] “Low Voltage Temperature Sensors,” 2010. [Online]. http://cdn.sparkfun.com/datasheets/Sensors/Temp/TMP35_36_37.pdf [Accessed: May 1, 2016].

[19] “Piezoelectric Buzzers,” 2011. [Online]. https://product.tdk.com/info/en/catalog/datasheets/ef532_ps.pdf [Accessed: May 1, 2016].

[20] “How to Install and Setup Eagle” 2016. [Online]. <https://learn.sparkfun.com/tutorials/how-to-install-and-setup-eagle/using-the-sparkfun-libraries> [Accessed: May 1, 2016].

[21] “Optimized fork of nRF24L01 for Arduino and Raspberry Pi” 2016. [Online]. https://product.tdk.com/info/en/catalog/datasheets/ef532_ps.pdf [Accessed: May 1, 2016].

Appendix A: Senior Project Analysis

1. Summary of Functional Requirements

The Car Safe Device functions as an array of sensors that take measurements based on its 4 primary inputs: heat, user proximity, decibel level of signals within the audible spectrum, and human baby presence. It uses the information it collects to notify a parent both audibly and visually if a child faces possible neglect. As its primary function, Car Safe works to protect children and parents by notifying a parent who walked away from a vehicle within which a child still sits. As secondary functions, the device monitors the ergonomic environment of a child seated in a car seat. Specifically, it measures the temperature and audio levels the child experiences and reports to the parent if the levels rise to a level not suitable for the comfort of the child.

2. Primary Constraints

The constraints surrounding Car Safe's development primarily include economic and social constraints. Limited funding exists for the project. Realistically the developmental costs, excluding labor costs, cannot exceed \$200. This limits the possibility of multiple device revisions during the developmental process making quality design practices imperative. Possible legal constraints also affect the project. Car Safe supports the goal of saving the lives of children and this opens up the possibility of harsh legal action made against Car Safe and its developers if it fails to operate as advertised. Refer to chapter 2 for a comprehensive list of the project's requirements.

3. Economic

Car Safe requires a moderate amount of capital when compared to larger scale projects. This includes human capital, financial capital, real capital, and natural capital. The human capital encompasses the electrical engineering student developing the product (myself), the Cal Poly administration advising the project, and the people used in testing Car Safe.

The expected financial capital required to build one Car Safe device sums to \$7,876.46. This value accounts for the predicted component and labor costs the project accrues. The labor costs of research and development accounts for most of the sum of \$6,024.90, representing the project's fixed cost. The other \$1,851.56 comes from the manufacturing labor and component costs and represents the variable cost of the project. This cost expectedly decreases as production rates increase [9]. As of now, the bulk of this financial cost falls upon Cal Poly via the electrical engineering department's Senior Project Fund, as well as the primary project developer: myself. This fails to meet the funds necessary for large-scale development and production. Investors would help meet the financial capital necessary in this case. Once Car Safe production increases to a profitable point, the investors and developers see a financial return. Whether production succeeds in increasing or not, Cal Poly still receives financial returns in the form of donations made by the Car Safe developer who receives a job in part based on his experience creating Car Safe.

The project requires access to an array of test equipment comprising of power supplies, oscilloscopes, function generators, and choke tubes to develop and characterize the operation of Car Safe. This equipment, along with the lab environments utilized developing the project, make up the real capital associated with Car Safe.

Finally, the project acquires environmental costs from the fossil fuels in the production of the power used to develop and operate Car Safe, and from the silicon used in the production of the components within the device itself. While the environmental costs remain minimal with the development of one device, the costs increase linearly with increased production.

4. Manufacturing on a Commercial Basis

The estimated number of Car Safe devices sold per year sits at a modest 100 units. This value stems from the little money spent on product advertisement and marketing. The per-unit cost of the device, not including labor, starts at \$182.49 and decreases with the production of more units. With a sale price of \$250 to the consumer, a projected profit of about \$6,751 minus labor cost appears. The production scale may rise if presented with quick market success. In this case, projected profits rise in relation to decreasing development costs per unit [9]. When considering the \$7,685.30 in labor costs associated with the project as fixed costs, a break-even point appears at 114 units according to the following equation:

$$X = \frac{\text{Fixed Cost}}{\text{Price per Unit} - \text{Variable Cost}} \quad \text{eq. 3}$$

5. Environmental

As stated in section 3, Car Safe contributes to the depletion of natural resources including fossil fuels, gold, and silicon; however, due to its low power usage and relatively limited number of components, it contributes a negligible amount. As a wireless device, the device possibly affects certain ecosystems including animals sensitive to radio frequencies; however, the frequencies it transmits exhibit very low power and provide limited range. Some ecosystems may experience indirect negative effects from the device if it transforms to e-waste, but the expected use model of the device does not promote Car Safe as a disposable product.

6. Manufacturability

Since Car Safe comes in contact with babies and young children, the device's housing features certain non-toxic materials. The use of non-toxic materials limits some manufacturing options including the use of certain toxic plastics. Also, the small size of the Bluetooth receiver portion of the device presents some manufacturing challenges such as custom receiver housings. At the current development stage, Car Safe's design requires assembly by hand. This places production limitations on the product. In order to increase both the speed and size of manufacturing, investments in manufacturing equipment must occur.

7. Sustainability

Car Safe's small physical size limits the possibility of hardware upgrades and repairs. Because of child safety concerns, the device's design restricts the user's ability to deconstruct Car Safe. A limitation of this design choice appears when entertaining the possibility of component failure. In this case, replacement of the entire device likely occurs, leaving the possibility that the user discards the device as e-waste. The implementation of software updates and bug fixes proves easier to provide. The design of the device can alter to include USB inputs that allow the device to connect to a PC to receive such an update.

This project directly impacts the sustainable use of resources in its limit use of electrical components. A slight negative impact presents itself as the components and materials used within Car Safe's creation appear in nature on a finite basis; however, the expected limited production of the device makes this impact nearly negligible. On the other hand, Car Safe positively affects sustainability in its implementation of rechargeable batteries. This protects from potential chemical waste from improper battery disposal accounting for a portion of the 20 million tons of e-waste produced each year, as according to iFixit [10].

8. Ethical

The Car Safe project aims to follow ethical engineering practices through adhering to the IEEE code of ethics and to Ethical Principlism. The project follows the IEEE code of ethics in its focus on child safety through the use of non-toxic materials [15]. Any potential safety dangers that present themselves in future design stages appears on the device packaging. The development of the device features no conflict of

interest since few third parties have direct financial stake in the device. All claims about Car Safe's functionality assert support from multiple tests and trials. Car Safe's developer accepts no forms of bribes for the project. Through the analysis of the social and ethical effects the project asserts, the developer worked to improve the understanding of the smart car seat technology and its consequences. The developer of the device claims qualifications to create such a device from his electrical engineering education and experiences at Cal Poly San Luis Obispo. The product development of Car Safe incurred honest criticism and critique through faculty advisement and potential customer surveys. The customer base surveyed and engineers consulted featured multiple ethnicities, ages, and genders. No instances of false or malicious action taken against another occurred in the development of Car Safe. A group of five engineers working on different senior projects formed with the purpose of assisting each other in part to meet the goals of the IEEE Code of Ethics. In these ways, Car Safe claims it ethically supports IEEE's code.

The project supports autonomy by allowing users the freedom of implementing Car Safe in a wide variety of car seats. The device aims to do good by protecting children from both neglect and discomfort while within an automobile. The product design incorporates considerations for child safety by not acting as a choking hazard and not utilizing toxic materials in its construction. This supports the ethical claim of nonmaleficence; however, in reality potential undesired effects of Car Safe can exist. As mentioned in sections 5 and 7, potential negative environmental byproducts exist as a result of Car Safe's use and development. While these environmental consequences do not occur as a part of the device's primary functionality, these harmful unintended results warrant consideration and concern during Car Safe's development. Finally, the device supports justice in keeping parents accountable for the actions they take regarding their children when within a car.

9. Health and Safety

The primary health and safety concerns of the project revolve around the wellbeing of the child in close proximity with the Car Safe device itself. The design of the product includes special precautions to insure the device features a non-toxic external housing that a child cannot choke on. The device itself must remain within reasonable handling temperatures to protect the child from potential burns or discomfort. A final health concern arises from the wireless transmitters present on the device.

During the device's design and manufacturing process, the health and safety of those building the device requires consideration. The potential for accidents arise when working with the electrical and manufacturing equipment used for the project's development. Because of this, the project developer established standards ensuring only properly trained hands utilize testing and manufacturing equipment during the project's development.

10. Social and Political

Child safety stands as a major concern worldwide. In the United States, and especially in California, lawmakers place a large importance on child safety within vehicles. As an example, a recent California law mandates children 2 years old and younger sit in rear facing car seats [2]. Car Safe helps parents assess the wellbeing and comfort of their children when they sit in car seats facing away from them.

The project developer stands as the primary beneficiary and stakeholder within the Car Safe project. With a successful project, he not only receives a good grade within Cal Poly's EE 460, EE 461, and EE 462, but also stands to gain financially from the possibility of selling Car Safe as a packaged project. On the other side, he also sees the largest backlash from a failed project. A failed project potentially affects the view future employers maintain on him impeding his ability to receive an ideal electrical engineering job. Cal Poly faculty and students also appear as major stakeholders in this project. Car Safe's future success relates directly to social status associated with the school.

On a more abstracted level, social and political organizations associated with child safety can consider themselves as stakeholders in this project. They benefit from the project if it brings about new standards in child vehicular safety. Little potential harm exists for these organizations, as a failed project cannot be released to the public meaning they lose nothing but simply miss an opportunity.

11. Development

For this project, I learned many project planning and design analysis tools including Gantt chart development and Monte Carlo analysis techniques. I also acquired the ability to analyze the economic considerations associated with product development. This included finding the projected fixed and variable costs of the project and analyzing economies of scale [8] [9]. The project's development required much research into present "smart" car seat technologies as well as short-range wireless communication technologies [3][4][12][13]. Finally, the project further developed my understanding of the processes used and considerations made when manufacturing an electrical product. It forced me to consider the manufacturing difficulties associated with making certain design choices. This greatly expands my knowledge and utility as an electrical engineer.

Appendix B: Hardware Schematics

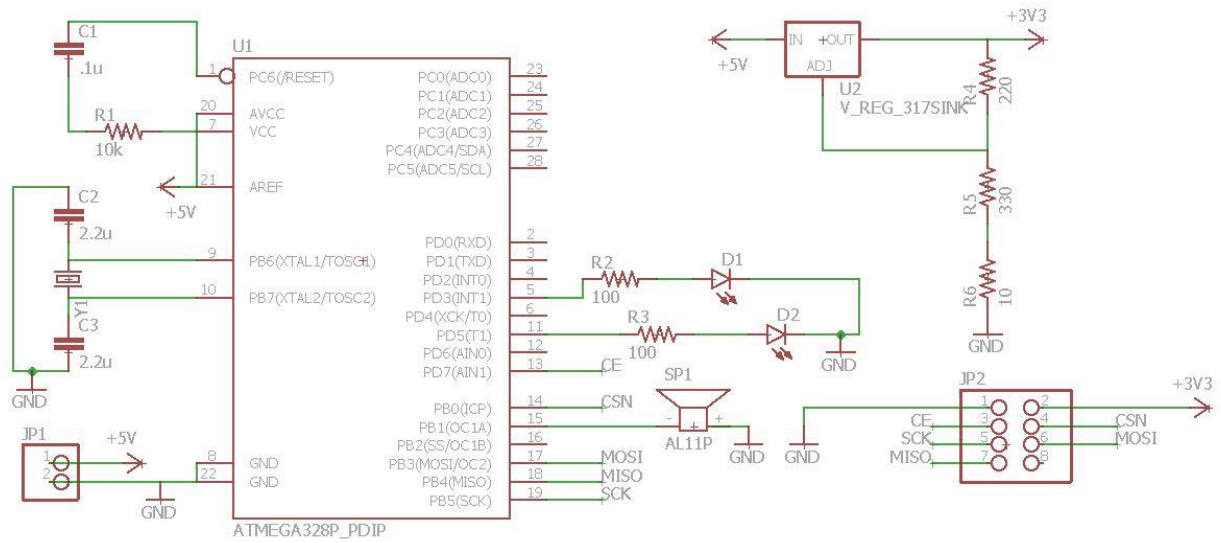


Figure B.1: Car Safe Receiver schematic

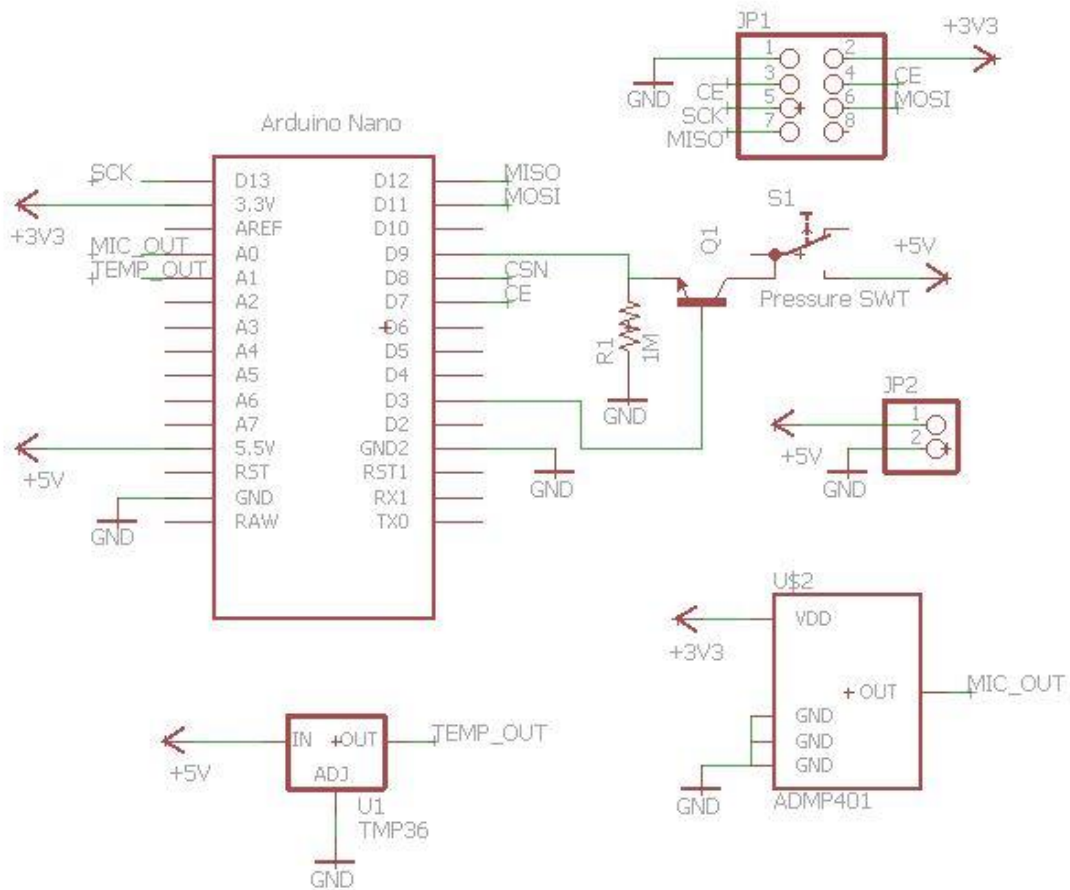
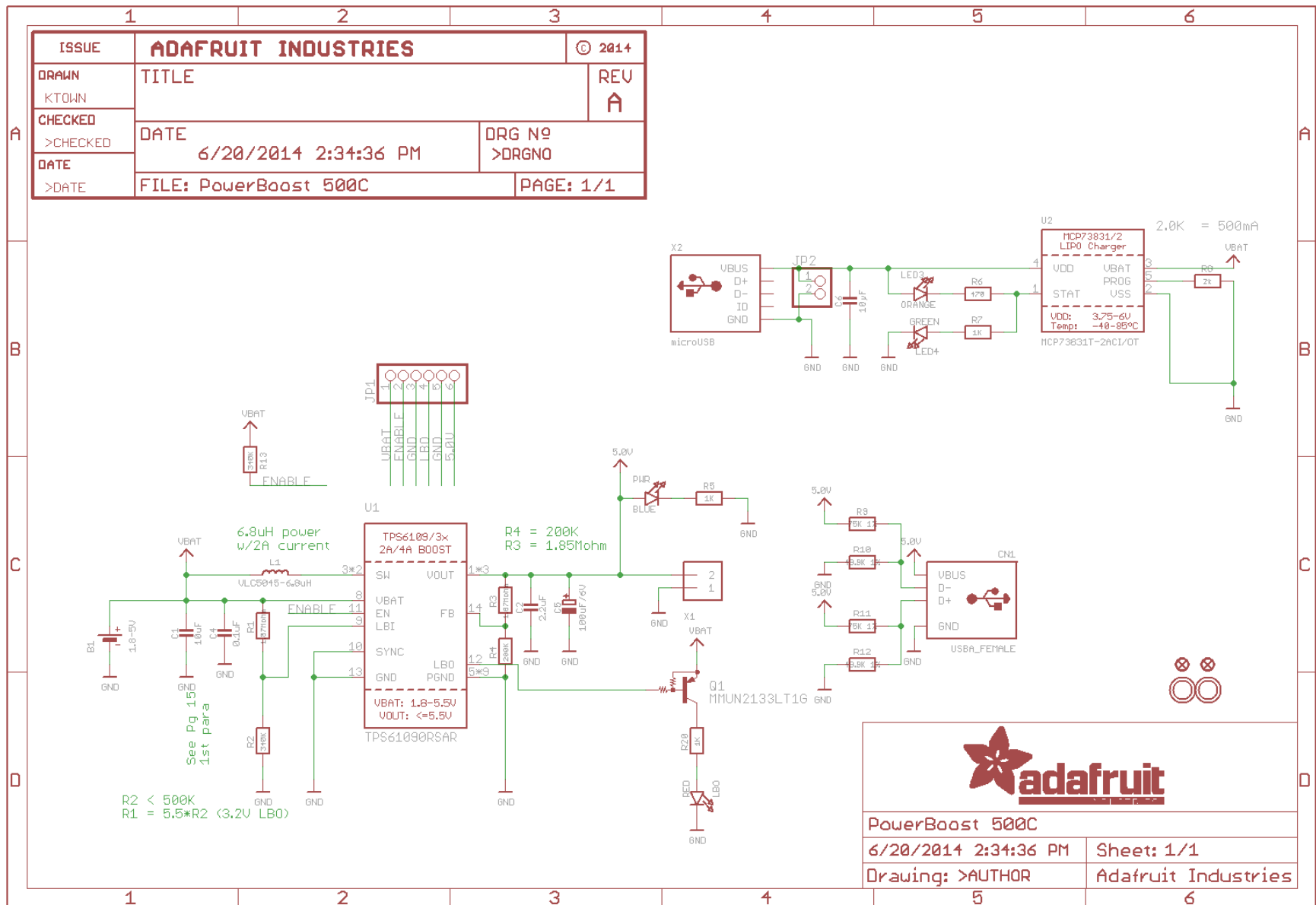


Figure B.2: Car Safe Transmitter schematic



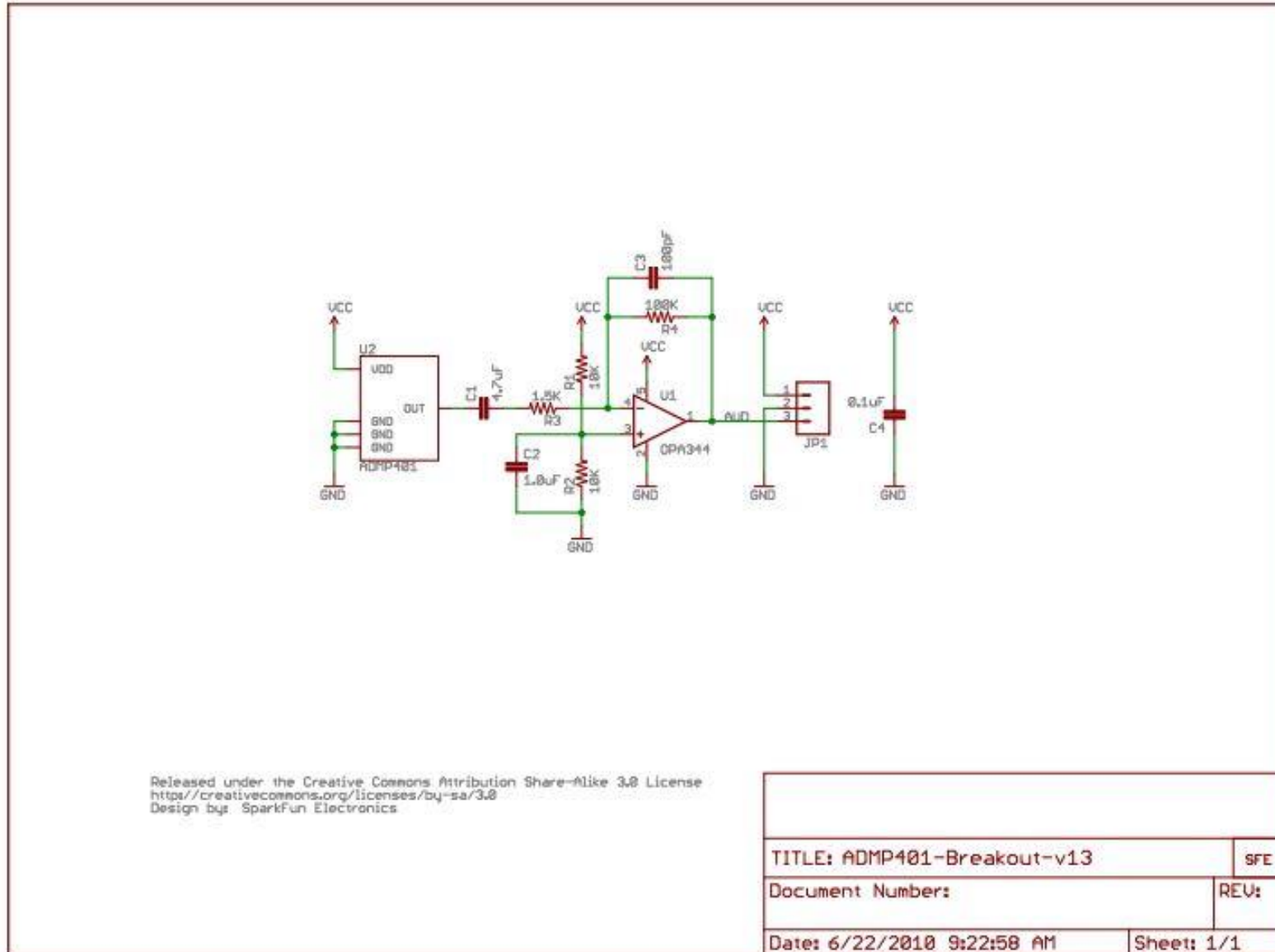


Figure B.4: SparkFun MEMS microphone breakout board schematic

Appendix C: Software Flowchart

Transmitter (Child Seat Side)

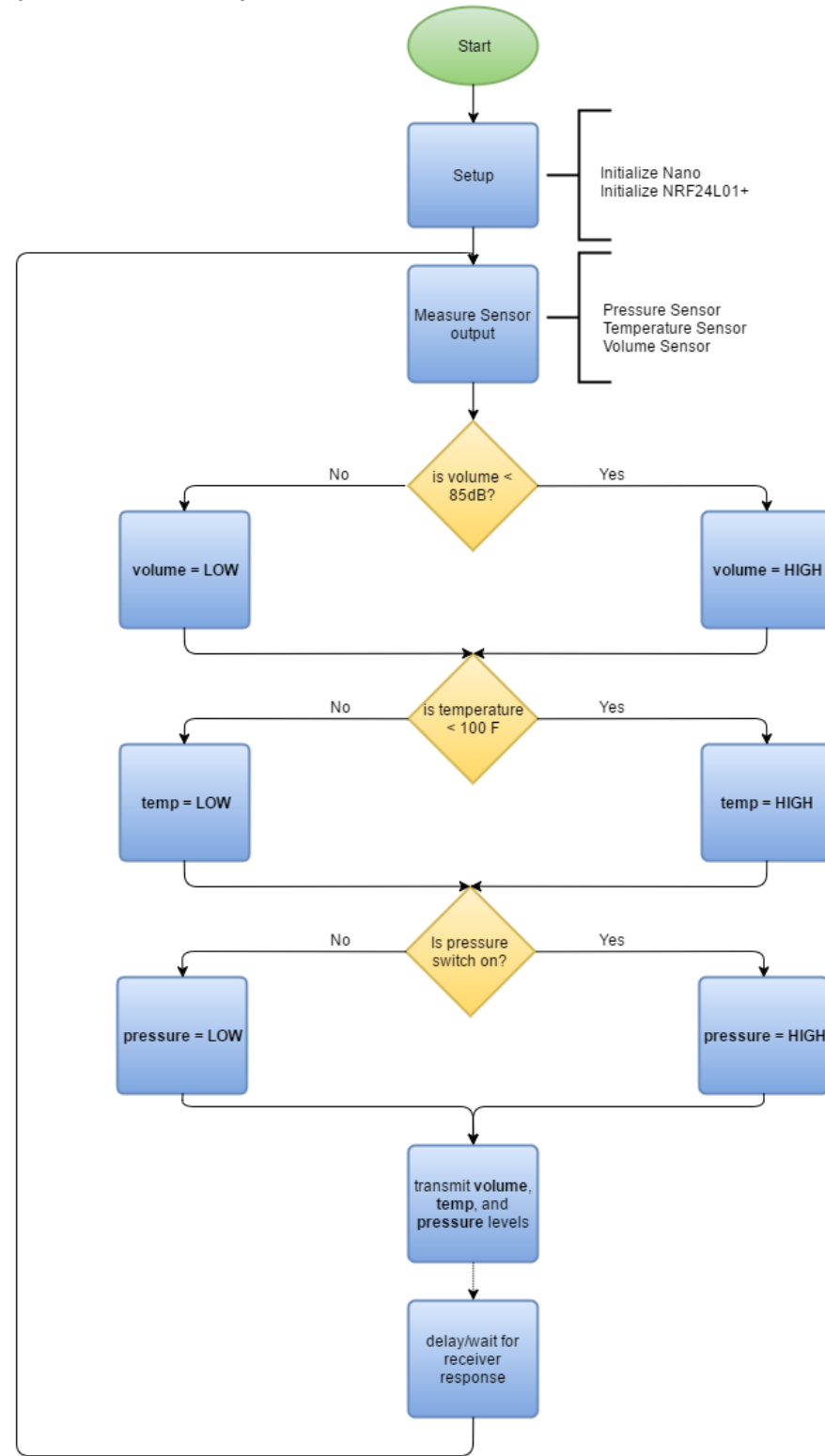


Figure C.1: Transmitter (Child Seat Side) program flowchart

Receiver (Parent Side)

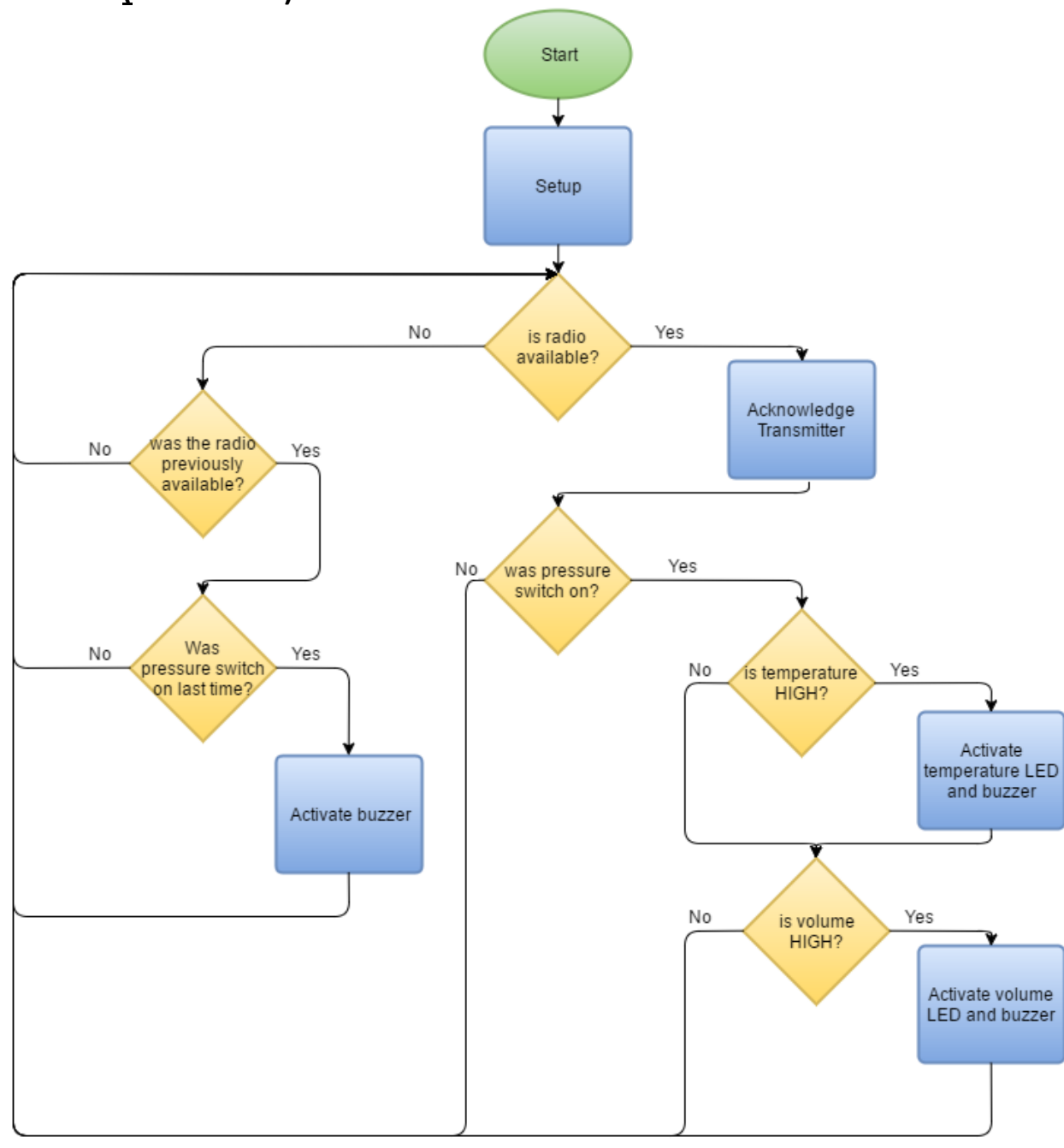


Figure C.2: Receiver (Parent Side) program flowchart

Appendix D: Software Code

Transmitter (Child Seat Side)

```
/* - nRF24L01 Radio Module: See http://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo
1 - GND
2 - VCC 3.3V !!! NOT 5V
3 - CE to Arduino pin 7
4 - CSN to Arduino pin 8
5 - SCK to Arduino pin 13
6 - MOSI to Arduino pin 11
7 - MISO to Arduino pin 12
8 - UNUSED
- V2.12 02/08/2016
- Uses the RF24 Library by TMRh20 and Maniacbug: https://github.com/TMRh20/RF24 (Download ZIP)
Questions: terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <SPI.h> // Comes with Arduino IDE
#include "RF24.h" // Download and Install (See above)
#include "printf.h" // Needed for "printDetails" Takes up some memory
/*-----( Declare Constants and Pin Numbers )-----*/
#define CE_PIN 7 // The pins to be used for CE and SN
#define CSN_PIN 8

#define MIC_OUT A0 // MEMS microphone output
#define TEMP_OUT A1 // Temperature output
#define LOUD_LED 3 //may remove
#define HOT_LED 5 //may remove
#define BUZZER_LED 6 //may remove
#define PRESSURE_SWITCH 9

/*-----( Declare objects )-----*/
/* Hardware configuration: Set up nRF24L01 radio on SPI bus plus (usually) pins 7 & 8 (Can be
changed) */
RF24 radio(CE_PIN, CSN_PIN);

/*-----( Declare Variables )-----*/
byte addresses[][6] = {"1Node", "2Node"}; // These will be the names of the "Pipes"

unsigned long timeNow; // Used to grab the current time, calculate delays
unsigned long started_waiting_at;
boolean timeout; // Timeout? True or False

//variables for sensors
int MicValue;
int TempValue;
int BabyPresent;
int TestNum;

//boolean hasHardware = false;
boolean hasHardware = true;

struct dataStruct {
    unsigned long _micros; // to save response times
    int Loud;
    int Hot;
    int Buzz;
    int BabyHere;
} myData; // This can be accessed in the form: myData.Volume_Loud etc.

void setup() /***** SETUP: RUNS ONCE *****/
{
    //sets pressure switch
    pinMode (PRESSURE_SWITCH, INPUT);
    pinMode (MIC_OUT, INPUT);
    pinMode (TEMP_OUT, INPUT);
}
```



```

//sets up serial
Serial.begin(115200); // MUST reset the Serial Monitor to 115200 (lower right of window )
// NOTE: The "F" in the print statements means "unchangable data; save in Flash Memory to
conserve SRAM"
Serial.println(F("Send sensor data by nRF24L01 radio to another Arduino"));
printf_begin(); // Needed for "printDetails" Takes up some memory

//initialize transmitter
radio.begin(); // Initialize the nRF24L01 Radio
radio.setChannel(108); // Above most WiFi frequencies
radio.setDataRate(RF24_250KBPS); // Fast enough.. Better range

// Set the Power Amplifier Level low to prevent power supply related issues since this is a
// getting_started sketch, and the likelihood of close proximity of the devices. RF24_PA_MAX is
default.
// PAlevel can be one of four levels: RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX
radio.setPALevel(RF24_PA_LOW);
// radio.setPALevel(RF24_PA_MAX);

// Open a writing and reading pipe on each radio, with opposite addresses
radio.openWritingPipe(addresses[0]);
radio.openReadingPipe(1, addresses[1]);

// Start the radio listening for data
radio.startListening();

// radio.printDetails(); // Uncomment to show LOTS of debugging information
} // --(end setup) ---

void loop() //***** LOOP: RUNS CONSTANTLY *****/
{
    radio.stopListening(); // First, stop listening so we can talk.

    SetData();

    //light up the Transmit LEDS for testing
    /*
    digitalWrite(LOUD_LED, myData.Loud);
    digitalWrite(HOT_LED, myData.Hot);
    digitalWrite(BUZZER_LED, myData.Buzz);
    */
    //serial stuff
    Serial.print(F(" Temp Value "));
    Serial.println(TempValue);
    Serial.print(F(" Temp Value "));
    Serial.println(MicValue);
    if(BabyPresent == 1)
    {
        Serial.println(F(" Pressure "));
    }
    else
    {
        Serial.println(F(" No Pressure "));
    }
}

myData._micros = micros(); // Send back for timing
Serial.print(F("Now sending - "));

    if (!radio.write( &myData, sizeof(myData) )) { // Send data, checking for error ("!"
means NOT)
        Serial.println(F("Transmit failed "));
    }

    radio.startListening(); // Now, continue listening

    started_waiting_at = micros(); // timeout period, get the current microseconds
    timeout = false; // variable to indicate if a response was received
    or not

```

```

while ( ! radio.available() ) { // While nothing is received
    if (micros() - started_waiting_at > 200000 ) { // If waited longer than 200ms,
        indicate timeout and exit while loop
        timeout = true;
        break;
    }
}

if ( timeout )
{ // Describe the results
    Serial.println(F("Response timed out - no Acknowledge."));
}
else
{
    // Grab the response, compare, and send to Serial Monitor
    radio.read( &myData, sizeof(myData) );
    timeNow = micros();

    // Show it
    Serial.print(F("Sent "));
    Serial.print(timeNow);
    Serial.print(F(", Got response "));
    Serial.print(myData._micros);
    Serial.print(F(", Round-trip delay "));
    Serial.print(timeNow - myData._micros);
    Serial.println(F(" microseconds "));
}

// Send again after delay. When working OK, change to something like 100
delay(100);

} //--(end main loop )---

/*-----( Declare User-written Functions )-----*/
void SetData()
{
    if (hasHardware) // Set in variables at top
    {
        /*****( Read the Sensor Levels )*****/
        MicValue = analogRead(MIC_OUT);
        TempValue = analogRead(TEMP_OUT);
        BabyPresent = digitalRead(PRESSURE_SWITCH);
    }
    else
    {
        MicValue = 400; // Send some known fake data
        TempValue = 112;
        if (TestNum > 20 && TestNum <=40)
        {
            BabyPresent = 0;
        }
        else if(TestNum > 40)
        {
            TestNum = 0;
        }
        else
        {
            BabyPresent = 1;
        }
        TestNum++;
    }
}

//initialize transmitter data
myData.Loud = LOW;
myData.Hot = LOW;
myData.Buzz = LOW;

//Set Baby presence
myData.BabyHere = BabyPresent;

```

```
//Compare to danger zones
if (MicValue >550)
{
    myData.Loud = HIGH;
    myData.Buzz = HIGH;
}
if (TempValue >170)
{
    myData.Hot = HIGH;
    myData.Buzz = HIGH;
}
}
//***** ( THE END )*****
```

Receiver (Parent Side)

```
/*
- CONNECTIONS:
- nRF24L01 Radio Module: See http://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo
1 - GND
2 - VCC 3.3V !!! NOT 5V
3 - CE to Arduino pin 7
4 - CSN to Arduino pin 8
5 - SCK to Arduino pin 13
6 - MOSI to Arduino pin 11
7 - MISO to Arduino pin 12
8 - UNUSED
- Volume led 3
- Temp led 5
- Piezo Buzzer 9

- Uses the RF24 Library by TMRh20 and Maniacbug: https://github.com/TMRh20/RF24 (Download ZIP)

/*-----( Import needed libraries )-----*/
#include <SPI.h>    // Comes with Arduino IDE
#include "RF24.h"   // Download and Install (See above)
#include "printf.h" // Needed for "printDetails" Takes up some memory
// NEED the SoftwareServo library installed
// http://playground.arduino.cc/uploads/ComponentLib/SoftwareServo.zip
// #include <SoftwareServo.h> // Regular Servo library creates timer conflict!
/*-----( Declare Constants and Pin Numbers )-----*/
#define CE_PIN      7    // The pins to be used for CE and SN
#define CSN_PIN     8
#define VOLUME_LED_PIN 3    // Pin Numbers for LED's and buzzer
#define TEMP_LED_PIN 5
#define BUZZER_PIN  9

/*-----( Declare objects )-----*/
/* Hardware configuration: Set up nRF24L01 radio on SPI bus plus (usually) pins 7 & 8 (Can be
changed) */
RF24 radio(CE_PIN, CSN_PIN);

/*-----( Declare Variables )-----*/
byte addresses[][6] = {"1Node", "2Node"}; // These will be the names of the "Pipes"

//boolean hasHardware = false;
boolean hasHardware = true;

boolean hasReceivedData = false;
boolean parentAway = false;
int babyInSeat = 0;
int noRadioCounter = 0;
int buzz = 0;

//for buzzer
const int songLength = 18;
char notes[] = "cdfda ag cdfdg gf "; // a space represents a rest
int beats[] = {1,1,1,1,1,1,4,4,2,1,1,1,1,1,1,4,4,2};
int tempo = 113;

//For data transmission
struct dataStruct {
    unsigned long _micros; // to save response times
    int Loud;
    int Hot;
    int Buzz;
    int BabyHere;
} myData; // This can be accessed in the form: myData.Volume_Loud etc.

void setup()    /***** SETUP: RUNS ONCE *****/
{
    //Serial setup
```

```

    Serial.begin(115200);    // MUST reset the Serial Monitor to 115200 (lower right of window )
    // NOTE: The "F" in the print statements means "unchangable data; save in Flash Memory to
conserve SRAM"
    Serial.println(F("Receive sensor data by nRF24L01 radio to another Arduino"));
    Serial.println(F("and control hardware if attached (Check 'hasHardware' variable)"));
    printf_begin(); // Needed for "printDetails" Takes up some memory

    //set pinmode
    pinMode(TEMP_LED_PIN, OUTPUT);    // sets the digital pin as output
    pinMode(VOLUME_LED_PIN, OUTPUT);  // sets the digital pin as output
    pinMode(BUZZER_PIN, OUTPUT);      // sets the digital pin as output

    InitializeReceiver();
} //--(end setup )---

void loop()    /***** LOOP: RUNS CONSTANTLY *****/
{
    if ( radio.available())
    {
        //has started receiving
        parentAway = false;
        hasReceivedData = true;
        noRadioCounter = 0;

        while (radio.available())    // While there is data ready to be retrieved from the receive
pipe
        {
            radio.read( &myData, sizeof(myData) );                // Get the data
        }

        radio.stopListening();                // First, stop listening so we can
transmit
        radio.write( &myData, sizeof(myData) );                // Send the received data back.
        radio.startListening();                // Now, resume listening so we catch the
next packets.

        //save babyInSeat state
        buzz = myData.Buzz;
        babyInSeat = myData.BabyHere;
        Serial.print(F("Packet Received - Sent response ")); // Print the received packet data
        Serial.println(myData._micros);
        Serial.print(F("Volume is "));
        Serial.println(myData.Loud);
        Serial.print(F(" Temp is "));
        Serial.println(myData.Hot);
        Serial.print(F(" Baby is "));
        Serial.println(babyInSeat);
        //activateBuzz();                //for test
    }
    /*radio unavailable*/
    else
    {
        noRadioCounter++;
        if(hasReceivedData && noRadioCounter > 10000)
        {
            hasReceivedData = false;
            if(babyInSeat == 1)
            {
                parentAway = true; //must set false somewhere
                Serial.println(F(" NO RADIO "));
                Serial.println(noRadioCounter);
            }
            else
            {
                parentAway = false;
            }
        }
    }
}

```

```

if (hasHardware)
{
    digitalWrite(VOLUME_LED_PIN, myData.Loud);
    digitalWrite(TEMP_LED_PIN, myData.Hot);
    //digitalWrite(BUZZER_PIN, myData.Buzz);
    if(parentAway)
    {
        radio.stopListening();
        activateBuzz();
        parentAway = false;
        InitializeReceiver();
    }
    else if((buzz != 0) && (babyInSeat == 1))
    {
        radio.stopListening();
        activateBuzz();
        buzz = 0;
        radio.startListening();
        Serial.println(F("out of loop"));
    }
} // END hasHardware
} //--(end main loop )---

/*-----( Declare User-written Functions )-----*/
void activateBuzz()
{
    int i, duration;

    for (i = 0; i < songLength; i++) // step through the song arrays
    {
        duration = beats[i] * tempo; // length of note/rest in ms
        Serial.print(F(" for loop "));
        Serial.println(i);
        if (notes[i] == ' ') // is this a rest?
        {
            delay(duration); // then pause for a moment
        }
        else // otherwise, play the note
        {
            tone(BUZZER_PIN, 4*frequency(notes[i]), duration);
            delay(duration); // wait for tone to finish
        }
        delay(tempo/10); // brief pause between notes
    }
}

int frequency(char note)
{
    // This function takes a note character (a-g), and returns the
    // corresponding frequency in Hz for the tone() function.

    int i;
    const int numNotes = 8; // number of notes we're storing

    // The following arrays hold the note characters and their
    // corresponding frequencies. The last "C" note is uppercase
    // to separate it from the first lowercase "c". If you want to
    // add more notes, you'll need to use unique characters.

    // For the "char" (character) type, we put single characters
    // in single quotes.

    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
    int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523};

    // Now we'll search through the letters in the array, and if
    // we find it, we'll return the frequency for that note.

    for (i = 0; i < numNotes; i++) // Step through the notes
    {
        if (names[i] == note) // Is this the one?

```

```

    {
        return(frequencies[i]);    // Yes! Return the frequency
    }
}
return(0); // We looked through everything and didn't find it,
           // but we still need to return a value, so return 0.
}

void InitializeReceiver()
{
    //initialize receiver
    radio.begin();    // Initialize the nRF24L01 Radio
    radio.setChannel(108); // 2.508 Ghz - Above most Wifi Channels
    radio.setDataRate(RF24_250KBPS); // Fast enough.. Better range

    // Set the Power Amplifier Level low to prevent power supply related issues since this is a
    // getting_started sketch, and the likelihood of close proximity of the devices. RF24_PA_MAX is
    // default.
    // PA Level can be one of four levels: RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX
    radio.setPALevel(RF24_PA_LOW);
    //    radio.setPALevel(RF24_PA_MAX);

    // Open a writing and reading pipe on each radio, with opposite addresses
    radio.openWritingPipe(addresses[1]);
    radio.openReadingPipe(1, addresses[0]);

    // Start the radio listening for data
    radio.startListening();
    //    radio.printDetails(); // Uncomment to show LOTS of debugging information
}
//***** ( THE END ) *****

```